1996

Automatic Identification of Icosahedral Virus Particles in Electron Micrographs

Ioana Maria Martin

Dan C. Marinescu

Report Number:
96-034

AUTOMATIC IDENTIFICATION OF ICOSAHEDRAL VIRUS PARTICLES IN ELECTRON MICROGRAPHS

Ioana M. Boier Martin
Dan C. Marinescu

CSD TR-96-034
June 1996

Automatic Identification of Icosahedral Virus Particles in Electron Micrographs

Ioana M. Boier Martin
Computer Sciences Department
Indiana University
South Bend, IN 46634

Dan C. Marinescu
Computer Sciences Department
Purdue University
West Lafayette, IN 47907

Abstract

Three-dimensional reconstruction of the structure of spherical viruses at high resolution (5 to 10Å) requires automatic identification of virus particles in electron micrographs. Existing methods are extremely sensitive to noise from various sources present in micrographs and the results reported are only modestly successful. In this paper we introduce the Crosspoint Method, a set of new algorithms and heuristics, and present a graphics package which supports both manual and automatic virus particle identification. The success rate of the Crosspoint Method is higher, there are considerably fewer false hits, and the running time is better than that of previously reported attempts.

1 Introduction

Image processing refers to any technique which alters and displays in a more tangible form the information contained in images. In the case of electron microscopy, it extends the scientist's ability to study images of biological structures because details that may be invisible to the naked eye can be clearly revealed. Noise from a variety of sources [2], such as variability in the specimen support film, impurities in the sample, irregularities at the atomic, molecular, or macromolecular level, defocus level, etc, appears in all micrographs to varying extents. Image processing methods identify and separate signal and noise components in the image. Clearer images allow better understanding of biological structure-function relationships. Correlation with X-ray diffraction, biochemical, genetic, immunological, and model building studies makes image processing a powerful tool for investigating the basis of molecular events in living systems. Since 1963 several techniques have been developed and are now routinely used in biological structure studies. One such technique is the reconstruction of a three-dimensional structure from a number of two-dimensional projections.

A micrograph containing a number of randomly oriented, well-preserved, symmetric particles can be processed to yield a three-dimensional structure by determining the relative positions of the symmetry elements. This approach has been most powerful when applied to particles with high symmetry such as spherical viruses because the high symmetry simplifies the determination of the positions of the symmetry elements in individual images and it also decreases the number of images required to reconstruct a three-dimensional structure at a given resolution [7]. Spherical viruses have icosahedral symmetry, that is, the symmetry of an icosahedron or that of a dodecahedron. Figure 1 shows the symmetry properties of an icosahedron. There are three types of rotations that bring it into self-coincidence. The symmetry elements corresponding to these rotations are twelve five-fold, twenty three-fold, and thirty two-fold axes of rotation.

The theoretical foundation for the three-dimensional reconstruction process is provided by the Projection Theorem which states that *the Fourier transform of the projected structure of a three-dimensional object is equivalent to a two-dimensional central section of the three-dimensional Fourier transform of the object, normal to the direction of projection* [5]. Indeed, the Fourier transform of the function $\rho(x, y, z)$ is:

$$F(X, Y, Z) = \int\int\int \rho(x, y, z) e^{2\pi i(xX+yY+zZ)} dx dy dz.$$

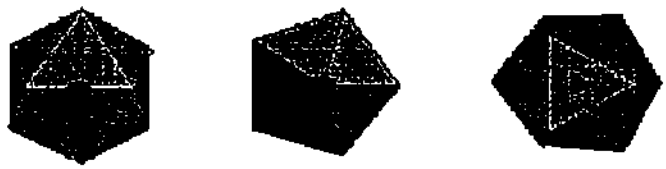

Figure 1: An icosahedron viewed along each of its symmetry axes.

The central section $Z = 0$ of the transform is given by:

$$F(X, Y, 0) = \int\int \sigma(x, y) e^{2\pi i(xX+yY)} dxdy, \text{ with } \sigma(x, y) = \int \rho(x, y, z) dz.$$

A typical reconstruction procedure includes the following steps. (a) *Image selection*: after an initial screening by eye (to discard obviously bad images), several micrographs are examined by optical diffraction and a small subset of the best images (in terms of optical quality and specimen preservation) is selected. The images in this subset are further processed. (b) *Densitometry* is the step in which the micrograph is digitized by converting optical densities into a numerical array. The devices that have commonly been used for densitometry are called densitometers. However, the use of CCD (charge coupled device) cameras for microndensitometry has gained popularity over the past few years as they have become more affordable. (c) *Boxing*: individual particle images are selected (boxed) and these constitute the different views used to reconstruct the three-dimensional structure under study. Their absolute orientations must be determined based on the symmetry the structure is assumed to possess. (d) The *Fourier transform* of each view of the particle provides a central section in the Fourier transform of the particle. By using data from several such views and by making use of the symmetry, a three-dimensional Fourier transform is filled up. (e) An *inverse Fourier transform* of the data collected in the previous step results in the desired, three-dimensional structure of the particle under consideration. Figure 2 shows a schematic representation of the reconstruction process (from [2]).

Boxing, the first step to be performed after the micrograph is digitized, has been traditionally considered a manual process, despite various attempts to automatize it. In this paper we introduce the Crosspoint Method for the detection of the positions of particle projections in electron micrographs. The complexity of the algorithm is linear in the number of pixels in the digitized micrograph and its performance is superior to all results reported in the literature when comparing both, the running time and, most importantly, the quality of the solution measured in terms of the ratio of the number of particles correctly identified and

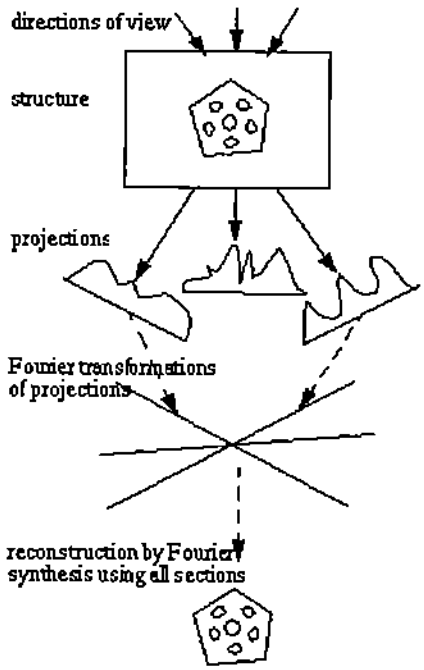

Figure 2: Schematic representation of the 3D reconstruction process.

the total number of particles. The number of false hits is also important because it affects the reconstruction process: the larger the number of false hits, the more noise is introduced in the reconstruction data.

Manual particle selection (e.g. using the mouse to point to and define each particle image) is not only time consuming and tedious, but also impractical in the case of high resolution reconstructions which require a large number of particle views. It is estimated that nearly 2000 particle images are necessary for a three-dimensional reconstruction to 10Å resolution of a virus approximately 1000Å in diameter [14].

The idea of using automatic particle selection methods is not new. Such methods have been proposed by M. van Heel in 1982 [9], by J. Frank and T. Wagenknecht in 1984 [6], by N. Olson and T. S. Baker in 1989 [12], and by P. Thuman-Commike and W. Chiu in 1995 [14]. All these methods use template-matching based on correlation to identify virus particles. The template is cross-correlated with the entire image and peaks in the resultant pattern identify the locations of regions in the image most similar to the reference. The method proposed in [14] is an extension of the basic template-matching procedure which involves extensive image preprocessing. The results reported show that the method is relatively slow and it generates a very large number of false particles which, if used, introduce a large amount of noise in the reconstruction. All the methods reported so far make use of Fourier transforms at one stage or another in the particle detection process and their complexity is therefore at least $O(n \times \log(n))$, where n is the number of pixels in the image.

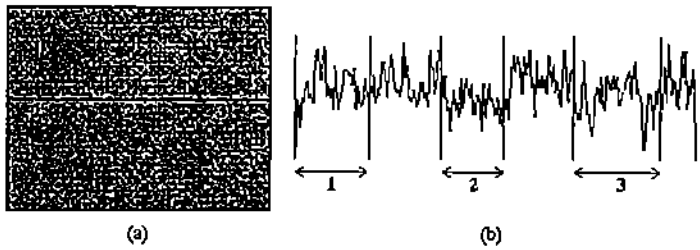

Figure 3: (a) Portion of a digitized micrograph. (b) The variation of the background intensity along a horizontal line.

Figure 4: The micrograph in Figure 3(a) after a Sobel transformation.

The main reason for which these as well as traditional image processing methods (e.g. edge and shape detection [3], [10], [11]) have failed is that they are extremely sensitive to the variation of the background intensity. Figure 3(a) shows a portion of a digitized micrograph which contains several particle images. The variation of the intensity along the horizontal line is depicted in Figure 3(b). It is clear that correlation and differentiation methods have little chance of success given the sharp transitions in intensity, both inside and outside particle projections. Figure 4 shows the result of applying a Sobel transformation to the micrograph shown in Figure 3(a). Sobel operators are a special case of gradient operators commonly used for edge detection in images [8]. As the example shows, the method failed to detect any circular edges in the digitized micrograph.

This paper is organized as follows. Section 2 describes the steps of the Crosspoint Method, including image enhancement techniques, algorithms for marking and clustering of pixels, and methods to improve the quality of the solution. Section 3 presents the experimental results obtained. An overview of a software package built around the Crosspoint Method is given in Section 4.

2 The Crosspoint Method for Automatic Particle Identification

The method we propose for automatic detection of the positions of particles in electron micrographs combines traditional image processing techniques with heuristics and a new algorithm for the detection of particle centers. It consists of four steps: (a) preprocessing, (b) particle identification, (c) clustering, and (d) postprocessing.

2.1 Preprocessing

In this phase the original digitized micrograph is transformed into an image more suitable for the algorithm which identifies the particle centers. Two traditional image processing techniques are used to enhance the micrograph: histogram equalization and averaging.

High resolution three-dimensional reconstruction requires closer to focus images. Such images have a low contrast and the role of histogram equalization is to enhance their contrast. Figures 5(a) and (b) show a micrograph before and after histogram equalization has been applied to it.

Averaging is motivated by the fact that the intensities of the pixels in the image are not characteristic to regions inside or outside a particle. A particular intensity value may occur inside some particle, as well as somewhere in the background. The human eye has the ability to identify the particles because low (dark) pixel intensities tend to predominate inside the particles and the high (light) values are scattered across these areas. The purpose of averaging is to extend the influence of the dark pixels inside the particles by smoothing out small regions of high intensity. Figure 5(c) shows a portion of the electron micrograph after histogram equalization and averaging. Since averaging is a smoothing operation, the image appears blurred. In this case a 10×10 mask was used for averaging.

2.2 Particle Identification

After the digitized micrograph has been preprocessed, an algorithm is used to identify pixels that belong to particles. This algorithm constitutes the core of the Crosspoint Method and it consists of marking pixels that are identified to be inside particle projections. The algorithm is described in the pseudo-code in Figure 6.

The algorithm takes as input the radius r of the particle. The image array is scanned

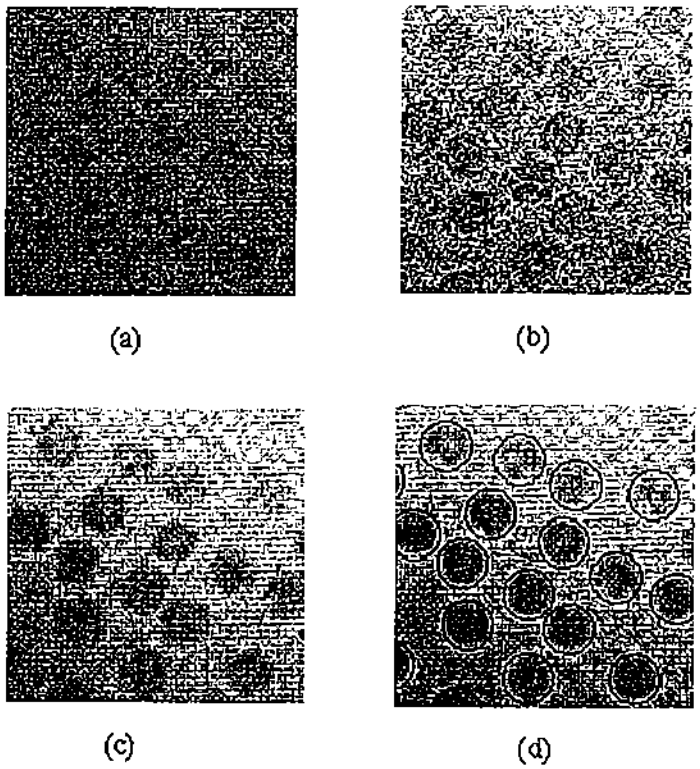

Figure 5: (a) Portion of a digitized micrograph. (b) Micrograph after histogram equalization. (c) Micrograph afetr equalization and averaging. (d) Final result of the Crosspoint Method.

horizontally and pairs of pixels at distance `r+1` are compared along the horizontal and vertical directions. The goal is to identify situations when one of the pixels in such a pair is outside and the other is inside a given particle. It is likely that the intensity values of pixels in such a pair differ significantly. If the two pixels are both either in the background or inside some particle, the likelihood of them having very different intensity values is small. The difference between the values of pixels in a pair is tested against a threshold value, `trsh`. The value of this threshold has been chosen heuristically, based on observations made on a number of digitized electron micrographs. The algorithm produces as output an array of 0s and 1s. The 1s correspond to pixels marked by the `markPixel()` routine and they represent pixels which are the most likely to be inside some particle.

2.3 Clustering

The third part of the Crosspoint Method consists of examining the marked pixels and detecting clusters of marked pixels. The clusters are used (a) to filter out marked pixels that are outside any particles by ignoring clusters that are too large or too small compared to

```
for each row j of pixels in the image do
    for each pixel i in row j do
        if (pixelValue([i, j]) - pixelValue([i+r+1,j]) ≥ trsh) then
            if (pixelValue([i+r+1, j]) - pixelValue([i+r+1,j+r+1]) ≥ trsh)
             then
                markPixel(i+r+1, j+r+1)
            else
                if (pixelValue([i+r+1, j]) - pixelValue([i+r+1,j+r+1]) ≤
                 -trsh) then
                    markPixel(i+r+1, j)
                endif
            endif
        else
            if (pixelValue([i, j]) - pixelValue([i+r+1,j]) ≤ -trsh) then
                if (pixelValue([i, j]) - pixelValue([i,j+r+1]) ≥ trsh) then
                    markPixel(i, j+r+1)
                else
                    if (pixelValue([i, j]) - pixelValue([i,j+r+1]) ≤ -trsh)
                     then
                        markPixel(i, j)
                    endif
                endif
            endif
        endif
    endfor
endfor
```

Figure 6: Pseudo-code describing the marking phase of the Crosspoint Method.

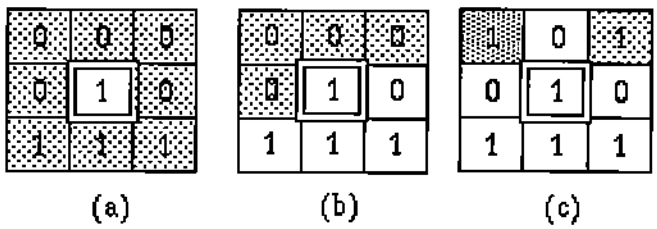

Figure 7: Various situations in the clustering algorithms: (a) eight neighbors of the current pixel are considered in the stack algorithm, (b) four neighbors of the current pixel are considered in the coloring algorithm, (c) two separate clusters must be merged in the coloring algorithm.

the area of the particle, and (b) to approximate the centers of each particle with the center of mass of the corresponding cluster. Two clustering algorithms are described here.

The *stack algorithm* detects connected components in a binary array using a stack. The array is scanned rowise, until the first 1 is encountered. Its coordinates are used to update the center of mass of the cluster currently being determined and also a counter that keeps the size of the cluster. All marked neighbors of the current pixel are pushed onto a stack (eight neighbors are considered; see Figure 7(a)). The next pixel to be processed in the same way is the one at the top of the stack. The processing of a cluster ends when the stack becomes empty. The horizontal scanning of the binary array then resumes, until all clusters have been detected. The pseudo-code for the stack algorithm is shown in Figure 8.

The *coloring algorithm* was suggested by M. J. Atallah [1]. It detects connected components in a binary array by "coloring" them in different colors. As in the stack algorithm, the array is scanned rowise and every time a marked pixel is encountered, it is either colored with a new color from a color array (if none of its neighbors is colored) or it receives the color of its neighbors. In this case only four neighbors are considered, as shown in Figure 7(b). A problematic situation is shown in Figure 7(c). At some point during the scanning process, two separate clusters, colored with two different colors become connected. In this case, the two clusters have to be "recolored" with the same color. A simple solution is to mark the two colors as being the same in the color array, rather than change the color of every pixel in one of the clusters. The center of mass and the size of the clusters can be computed on-the-fly, as the scanning progresses. The pseudo-code for the coloring algorithm is shown in Figure 9.

```
for j = 1 to number-of-rows do
    for i = 1 to number-of-columns do
        if (pixel (i, j) is marked) then
            processPixel(i, j)
            while (Stack is not empty) do
                pop item (i1, j1) from Stack
                processPixel(i1, j1)
            endwhile
            if (clusterSize within limits) then
                accept cluster
            else
                ignore cluster
            endif
        endif
    endfor
endfor


processPixel (i, j)
begin
    push all marked neighbors of (i, j) which have not yet been considered
     onto Stack
    update clusterSize
    update centerOfMass
    unmark pixel (i, j)
end
```

Figure 8: Pseudo-code describing the stack algorithm for clustering.

```
for j = 1 to number-of-rows do
    for i = 1 to number-of-columns do
        if (pixel (i, j) is marked) then
            C = getColor(i, j)
            color pixel (i, j) with color C
            update clusterSize[C]
            update centerOfMass[C]
        endif
    endfor
endfor


getColor (i, j)
begin
    if (pixel (i, j) has no colored neighbors) then
        C = new color
        return C
    endif
    if (pixel (i, j) has all pixels colored C) then
        return C
    endif
    if (pixel (i, j) has neighbors of two different colors, C1 and C2) then
        merge clusters C1 and C2
        C1 = C2
        return C1
    endif
end
```

Figure 9: Pseudo-code describing the coloring algorithm for clustering.

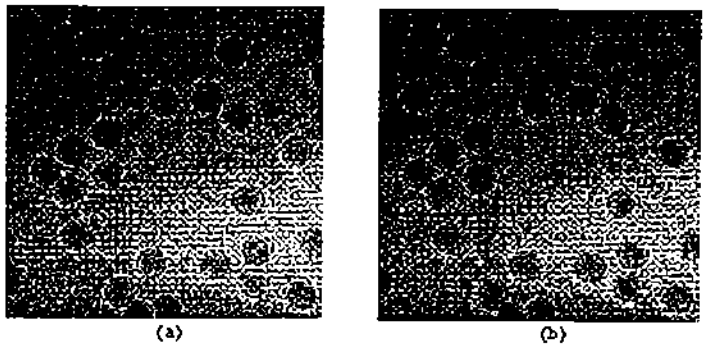

Figure 10: Sensitivity of the Crosspoint Method to changes in radius in the case of the REO-18 micrograph: (a) r = 41, (b) r = 42

2.4 Postprocessing

Three types of errors may occur when attempting to automatically identify the positions of particles: (a) missed particles, (b) separate particles clustered together, and (c) "ghosts" which do not correspond to real particles.

There are a number of reasons for which a particle may be missed. Usually, the pixel intensities inside and around such a particle fail to obey the general thresholding rule, used to build the clusters. The cluster corresponding to such a particle is either too large or too small compared to the particle's area and it is discarded during the clustering step. Changing the radius of the particle may affect the identification of such particles, although the overall results are not very sensitive to small variations in the value of the radius.

Figure 10 shows the solutions obtained for two consecutive values of the radius in the case of the REO-18 micrograph. Our studies show that the quality of the solution does not change substantially for small variations of r. Since r is defined interactively, the algorithm should be insensitive to small variations of the particle radius. The disadvantage is that the method cannot discriminate between virus particles of different sizes that appear in the same micrograph.

Particles that are spatially close to each other may be incorrectly identified as one cluster. In this case, a single particle is incorrectly identified somewhere between the particles clustered together. This problem can be solved by using a *thinning* procedure. This procedure is based on the observation that, in most cases, such clustered particles are connected by a thin bridge of marked pixels. By removing some of the outmost layers of pixels from the

Figure 11: Disconnecting particles by thinning: (a) particle identification without thinning, (b) particle identification with thinning.

cluster, the bridge is disconnected, and the two particles may be separately identified. There is a trade-off between how many layers one can remove from the cluster and the chances of detecting the two separate particles after thinning. If the number of layers that need to be removed in order to break the bridge is large, the size of the two clusters may decrease considerably and it may not satisfy the size requirements a cluster has to meet in order to be considered as representing a particle. Therefore, it is a good idea to adjust the cluster limits after thinning, by a number equal to the total perimeter of the layers which have been removed. Figure 11 shows a situation in which two particles clustered together are correctly identified after thinning.

The thinning algorithm takes as input the number of layers to be removed and it proceeds with the removal after each cluster has been identified. A marked pixel is removed from a cluster if it is a border pixel, that is, if not all of its neighbors are marked pixels. Figure 5(d) shows the final result of applying the Crosspoint algorithm to the digitized micrograph shown in Figure 5(a). The test used to reduce the number of "ghost" particles consists of a comparison between the average intensity inside each detected particle and the average intensity on a circle immediately outside it. If the two values are very similar, then chances are that the particle is a "ghost" particle, lying somewhere in the background. For a true particle these two average values should be quite different. The threshold used for the comparison is chosen heuristically (it is usually the same as the threshold used for particle detection).

3 Experimental Results

The performance of the automatic particle location algorithm can be measured in terms of the running time and the quality of the solution. Algorithm efficiency is important when a large number of micrographs are processed. As the size of the images increases, from say 1000×1000 to 6000×6000 pixels, the efficiency of the algorithms and the amount of memory available become critical. Dividing the image into subimages as suggested in [14] complicates the algorithms and may lead to a substantial increase in the running time. Manual procedures are not only tedious, but also very slow and prone to errors. Automatic procedures should be much faster, particularly when used in conjunction with automatic microscope control. As mentioned before, previous algorithms for automatic particle selection are of complexity $O(n \times \log(n))$, where n is the number of pixels in the digitized electron micrograph. The Crosspoint Method has complexity O(n) and, according to our estimates, the timing results are approximately an order of magnitude better than those reported in the literature. It is, however, difficult to make precise comparisons because of differences in the speed of the processors used and in the amount of memory available.

Table 1 gives the time in seconds for each of the main processing steps of the Crosspoint Method, applied to images of various sizes and particle content. These results have been obtained on a Silicon Graphics workstation, equipped with a 200 MHZ MIPS R4400 processor, and 64 MB main memory. Information about the biological content of the images used for testing is given in Appendix A.

Image	Size (pixels)	Histogram Equalization (s)	Averaging (s)	Particle Identification and Postprocessing (s)	Total (s)
PHIX-A	901 × 901	1.5	5	4	10.5
PHIX-B	901 × 901	1.5	5	4	10.5
BMV	901 × 901	1.5	5	4	10.5
REO-18	901 × 901	1.5	5	4	10.5
T1LHC	2560 × 2000	27	48	42	117

Table 1. Timing results for the Crosspoint Method.

Correct identification of particles has three aspects, corresponding to the three types of errors described in §2.4. To appreciate the performance of an automatic particle detection

algorithm, one needs to know how many of the particles present in the image have been identified, how many have been missed, and how many "ghost" particles have been introduced. Ideally, all particles in the image should be identified and no "ghost" particles should be introduced. Missed particles are not a serious problem, as long as there are not too many of them. "Ghost" particles, however, are more harmful because they introduce noise in the reconstruction. Table 2 gives information about the number of particles correctly identified by the Crosspoint Method. Note the very small percentages of missed and "ghost" particles. In [14] P. Thuman-Commike reports percentages of "ghost" particles between 30% and 55% which, in our opinion, makes the overall solution unusable for three-dimensional reconstruction unless some further processing is applied to remove these particles.

Image	Total # of particle images	Correctly Identified	Missed	"Ghost"
PHIX-A	71	64 (90%)	7 (10%)	2 (3%)
PHIX-B	47	42 (89%)	5 (11%)	2 (4%)
BMV	130	120 (92%)	10 (8%)	0 (0%)
REO-18	21	21 (100%)	0 (0%)	0 (0%)
T1LHC	100	95 (95%)	5 (5%)	1 (1%)

Table 2. Quality of the results: the number of particles correctly and incorrectly identified by the Crosspoint Method.

Figures 12 through 16 show the result of the Crosspoint Method applied to each of the images listed in Table 2. Note that the border particles are not considered in the Crosspoint algorithm and they have not been included in the previous tables. Such particles are clipped and are not suitable for use in the reconstruction process. The percentage of "ghost" images is calculated from the total number of identified image regions. The percentage of the correctly identified and missed particle images is calculated from the total number of particle images in the micrograph.

4 A Software Package for Particle Identification

EMMA is an interactive software package built around the Crosspoint algorithms. In addition to automatic particle selection, it includes capabilities that allow the electron microscopist to perform various traditional image processing transforms on the digitized micrographs, to manually select, deselect, and extract individual particles, and to store particles

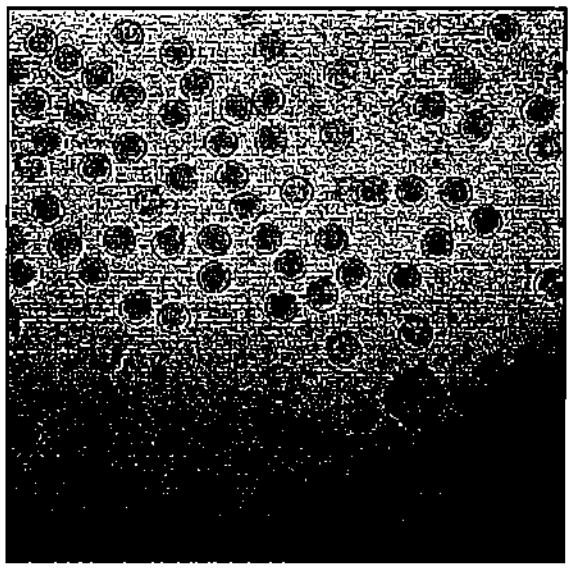

Figure 12: The result of the Crosspoint Method applied to the PHIX-A micrograph.

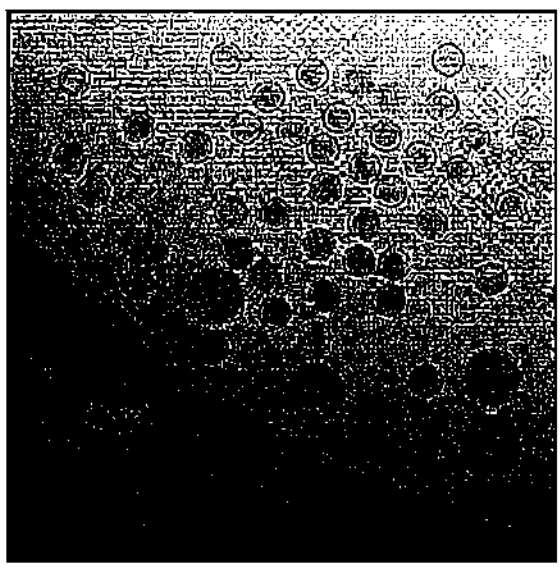

Figure 13: The result of the Crosspoint Method applied to the PHIX-B micrograph.

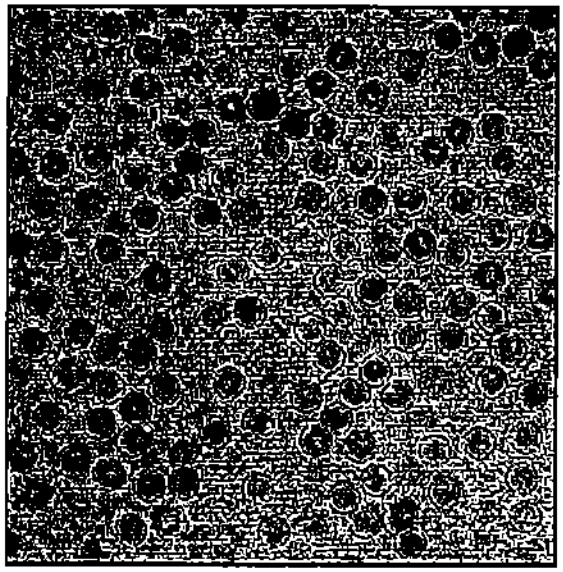

Figure 14: The result of the Crosspoint Method applied to the BMV micrograph.

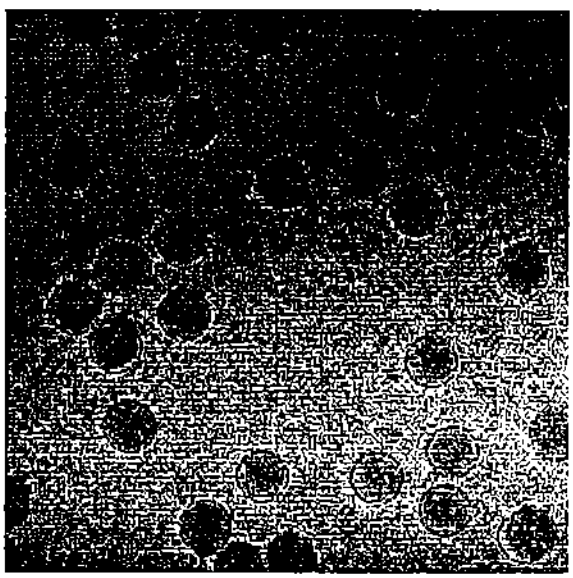

Figure 15: The result of the Crosspoint Method applied to the REO-18 micrograph.

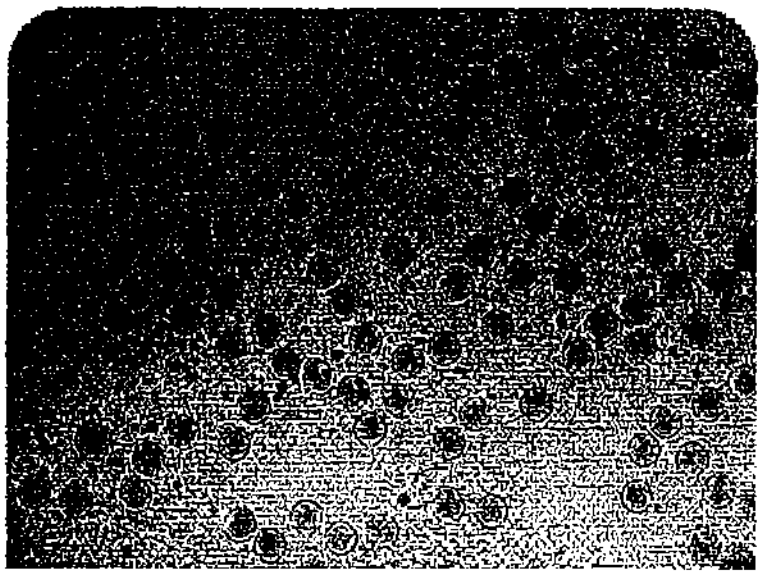

Figure 16: The result of the Crosspoint Method applied to the T1LHC micrograph.

into files. EMMA is built in X-Windows and Motif [15] and consists of approximately 15,000 lines of code.

The skeleton of the user interface was generated using BX [4]. Figure 17 depicts a snapshot of the screen during an EMMA session. The main, colormap, and histogram windows are shown. The main window is the only one that stays on the screen for the entire duration of an EMMA session. The other windows are displayed as needed. The main window consists of a menu, a drawing area, and a command area. The menu has several submenus, for I/O functions, for automatic particle selection, and for image processing. The drawing area can be scrolled horizontally and vertically to enable the user to view the entire image when this exceeds the size of the drawing window. Initially, the image is scaled so that no scrolling is needed. Buttons in the command area allow the user to zoom and shrink the image. The command area also contains widgets for manual particle selection. The colormap editor serves to modify the mapping of optical density values to colors.

Once an image is displayed and the radius of the particles to be found is defined, the user can invoke the Crosspoint Method to determine the positions of the particles in the image. Individual steps of the method (histogram equalization, averaging, and particle identification) can also be invoked separately. A toggle button in the command area allows the user to turn the displaying of clusters on and off. Selected particles may be manually deselected.

Figure 17: Snapshot of the screen during an EMMA session.

Manual particle selection has been implemented via polygonal and circular rubberbanding. The user may define a clipping boundary in the form of a polygon or a circle using the mouse. This boundary may or may not retain its shape from one selection to the next, depending on the setting the user chooses from the command area. The interior content of the rubberbanding area can be extracted in a rectangular window and padded with an average color intensity. Extracted regions can be saved into files and subsequently displayed.

Traditional image processing transforms may be performed in EMMA. The transforms supported are: histogram equalization, averaging, Sobel and Laplace gradient methods, high-boost filtering, colormap modification, and the Hough transform. Each of these transforms is applied to the image currently displayed in the drawing area. This allows for an easy composition of transformations in different orders.

5 Conclusions

In this paper we describe algorithms and heuristics for automatic identification of the positions of virus particles in electron micrographs and we present the results obtained. There are two major problems that have made automatic particle identification a difficult problem: noise in the micrographs and variations in the background intensity. Traditional image

processing techniques such as edge or shape detection yield poor results when applied to digitized electron micrographs. Specific techniques based on correlation methods have also been proposed [14]. Such methods require a considerable amount of computing time, their complexity is at least $O(n \times \log(n))$ (n is the number of pixels in the image) and the quality of the solution is not satisfactory. The Crosspoint Method proposed in this paper is a linear algorithm for solving the automatic particle detection problem and the results obtained for micrographs of various sizes and particle content and with different defocus levels have proved much better than those reported in the literature. The method combines traditional image processing techniques such as histogram equalization and averaging, with heuristics and a new algorithm for marking pixels inside particle projections. The marked pixels are then used to form clusters and to approximate the positions of the centers of the particle projections. Postprocessing is used to improve the quality of the solution by reducing the number of false hits.

Acknowledgments

We would like to thank Mikhail J. Atallah for suggesting the coloring algorithm for clustering and Timothy S. Baker for his constructive criticism and advice and for providing the data sets to test our programs. This research has been partially supported by the National Science Foundation grants BIR-9301210 and MCB-9527131, by a grant from the Intel Corporation, a grant from the Purdue Research Foundation, and by the Scalable I/O Initiative.

References

[1] M. J. Atallah, private communications.

[2] T. S. Baker, private communications and class notes.

[3] D. H. Ballard, Generalizing the Hough Transform to Detect Arbitrary Shapes, *Pattern Recognition*, Vol. 13, Number 2, pages 111–122, 1981.

[4] ***, *Builder Xcessory User's Guide*, version 3.0, Integrated Computer Solutions, Inc., 1993.

[5] R. A. Crowther, D.J. DeRosier, and A. Klug, The Reconstruction of a Three-Dimensional Structure from Projections and Its Applications to electron Microscopy, *Proceedings of the Royal Society London*, A 317, pages 319–340, 1970.

[6] J. Frank and T. Wagenknecht, Automatic Selection of Molecular Images from Electron Microscopy, *Ultramicroscopy*, Number 12, pages 169–175, 1984.

[7] S. D. Fuller, S. J. Butcher, R. H. Cheng, and T.S. Baker, Three-Dimensional Reconstruction of Icosahedral Particles - The Uncommon Line, *Journal of Structural Biology*, Vol. 116, Number 1, pages 48–55, 1996.

[8] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, Addison-Wesley Publishing Company, Inc., 1993.

[9] M. van Heel, Detection of Objects in Quantum-Noise-Limited Images, *Ultramicroscopy*, Number 8, pages 331–342, 1982.

[10] P.V.C. Hough, Methods and Means for Recognizing Complex Patterns, U.S. Patent 306954, 1962.

[11] C. Kimme, D. Ballard, and J. Sklansky, Finding Circles by an Array of Accumulators, *Communications of the ACM*, Vol. 18, Number 2, pages 120–122, 1975.

[12] N. H. Olson and T. S. Baker, Magnification Calibration and the Determination of Spherical Virus Diameters Using Cryo-Microscopy, *Ultramicroscopy*, Number 30, pages 281–298, 1989.

[13] N. H. Olson, private communications.

[14] P. Thuman-Commike and W. Chiu, Automatic Detection of Spherical Particles from Spot-Scan Electron Microscopy Images, *JMSA*, Vol. 1, Number 5, pages 191–201, 1995.

[15] D.A. Young, *The X Window System Programming and Applications with Xt*, the OSF/Motif Edition, Prentice-Hall, Inc., 1990.

Appendix A: Electron Micrographs Used for Testing the Crosspoint Method

PHIX-A and PHIX-B are images of Bacteriophage Phi X 174 virus particles. This is a virus that infects the bacteria Eschericha Coli which is found in the digestive system of all higher animals (humans included). Phi X 174 has been studied because it is a small (310Å diameter) virus that contains DNA and it is one of the first viruses whose entire genome was determined. Because of that it, serves as a "biological standard" in virus studies. In the images presented, the Phi X 174 particles have been mixed with polyoma virus particles. Polyoma is the larger virus (495Å) and its diameter was fairly accurately determined by X-ray diffraction. It serves as a magnification standard for the smaller Phi X 174. Polyoma infects recently born mice and it is heavily studied because similar viruses cause warts and possibly cervical cancer in humans [13].

BMV contains particles of the bromegrass mosaic virus. This is a plant virus that infects bromegrass. The virus particle is 268Å in diameter. Scientists have been able to purify the gene that actually produces the protein of the outer shell of the virus.

REO-18 and T1LHC are intermediate particles of reovirus. This virus infects the respiratory and digestive pathways of mammals. It consists of a complex system of many proteins. Its diameter is about 400Å.